UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/050,358	01/16/2002	Nicolai Kosche	188186/US/2	5734

66083 7590 05/28/2008
SUN MICROSYSTEMS, INC. c/o DORSEY & WHITNEY, LLP
370 SEVENTEENTH ST.
SUITE 4700
DENVER, CO 80202

EXAMINER	
WANG, RONGFA PHILIP	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
05/28/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/050,358	**Applicant(s)** KOSCHE ET AL.	
	Examiner PHILIP WANG	**Art Unit** 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 February 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3,5-31,37,38,46,47,49-54,59-66,68 and 69* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3,5-31,37,38,46,47,49-54,59-66,68 and 69* is/are rejected.

7)☒ Claim(s) *19* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response to amendment filed on 2/29/2008.

2. Per Applicant's request, claims 1, 3, 5, 19, 46, and 59have been amended.

3. The objection to the Specification has been withdrawn in view of the Applicant's amendment to the Specification.

4. Claims 1, 3, 5-31, 37-38, 46-47, 49-54 and 59-66, and 68-69 remain pending.

Claim Objections

5. Claim 19 is objected to because of the following informalities: it appears there should be a "," after "is not encounter" in the limitation "when the ambiguity creating location is not encountered associating the execution event with the particular operation". Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

6. Claims 1, 3, 5-18, 19-31, 37-38, 46-47, 49, 59-60, 62-66, 68-69 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Claim 63 depends on claim 1,claim 60 depends on claim 59 and claims 64-69 depend on claim 19. The independent claims

1, 19, 46, and 59 include the limitation of “wherein the displacement is based on, at least in part, on a type of operation appropriate to have triggered the execution event”. Upon reviewing the Applicant’s specification, [1030] (or similarly [1038]) --

> “Beginning at the coinciding point for a particular execution event detection, collection steps back through the original execution sequence in an attempt to find a preceding target instruction (i.e., an instruction of appropriate type to have triggered the detected event). In some realizations, an expected displacement is reasonably fixed and may place an upper bound on the backtracking.”

The above disclosure appears to suggest the backtracking attempts to find a preceding target instruction of appropriate type to have triggered the detected event. That means the type of instruction that has triggered the detected event is yet to be determined. However, the claim language appears to be able to determine a displacement for backtracking based on a type that appears to be unknown when such determination is made. Further the specification appears to suggest an expected displacement is reasonably fixed or there is an upper bound on the backtracking. It does not appear to disclose any determination of displacement based on a type of operation. Therefore, the examiner considers there is no disclosure of such limitation.

Further claim 46-47, and 49, include the limitation of "associating the execution event with one or more of the ambiguity creating locations..." It appears that the current invention is related to associating the execution event with a target instruction, instead of an ambiguity creating locations.

Claim 62 depends on claim 3 and suffers the same deficiency.

7. Claims 1, 3, 5-18, 19-31, 37-38, 46-47, 49, 59-60, 62-66, 68-69 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. The independent claims 1, 19, 46, and 59 include the limitation of "wherein the displacement is based on, at least in part, on a type of operation appropriate to have triggered the execution event". Upon reviewing the Applicant's specification, [1030] (or similarly [1038]) --

"Beginning at the coinciding point for a particular execution event detection, collection steps back through the original execution sequence in an attempt to find a preceding target instruction (i.e., an instruction of appropriate type to have triggered the detected event). In some realizations, an expected displacement is reasonably fixed and may place an upper bound on the backtracking."

The above disclosure appears to suggest the backtracking attempts to find a preceding target instruction of appropriate type to have triggered the detected event. That means the type of instruction that has triggered the detected event is yet to be determined. However, the claim language appears to be able to determine a displacement for backtracking based on a type that appears to be unknown when such determination is made. Further the specification appears to suggest an expected displacement is reasonably fixed or there is an upper bound on the backtracking. It does not appear to disclose any determination of displacement based on a type of operation. Therefore, the examiner considers there is lack of disclosure in the specification in such a way as to enable one skilled in the art to make or use the invention.

Further claims 46-47, and 49, include the limitation of "associating the execution event with one or more of the ambiguity creating locations..." It appears that the current invention is related to associating the execution event with a target instruction, instead of an ambiguity creating locations.

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. Claims 1, 3, 5-18, 19-31, 37-38, 46, 47, 49, 60, 62-66, 68 are rejected under 35 U.S.C. 112, second paragraph, as failing to set forth the subject matter which applicant(s) regard as their invention. Evidence that claims 1, 19 and 59 fail(s) to correspond in scope with that which applicant(s) regard as the invention can be found in [1022] of page 7, lines 4-6, "Absent the ambiguity creating location, backtracking from the miss detection to load instruction is straightforward". The purpose of the instant invention is to perform backtracking while in a skid region containing ambiguity creating locations, which is contrary to what the claims language of independent claims 1, 19 and 59. Independent claims 1, 19, and 59 added the limitation of (or similar limitation as in claims 19, 59) "when the creating is not encountered, associating the execution event with the preceding operation;". Per Applicant's remark, the invention appears to be able to perform backtracking and associating an event to a target instruction regardless of if an ambiguity creating location is encountered during backtracking. However, at least the claim language of the independent claims does not appear to include any limitation related to how to associate an event to a target instruction while encountering an ambiguity creating location during backtracking. Therefore all dependent claims of claims 1, 19 and 59 suffer the same deficiency.

Further claims 46-47, and 49, include the limitation of "associating the execution event with one or more of the ambiguity creating locations..." It appears that the current invention is related to associating the execution event with a target instruction, instead of an ambiguity creating locations.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

9. Claims 1, 3, 5-15, 19-31, 37-38, 46, 47, 49, 59-60, 62 are rejected under 35 U.S.C. 102(b) as being anticipated by Anderson et al. (US Patent No. 5,964,867).

As per claim 1, Anderson et al. disclose

During machine code generation of the code (c23: 24-31, "Many compiler optimizations …either through instrumentation or simulation…"),

identifying an ambiguity creating location in the code, the ambiguity creation location being an operation reachable from a plurality of program execution paths (col. 1, 35-40, shows using instrumentation for profiling code; col. 23, line 55-col. 24, line 31, for example, col. 24, line 9, "...ambiguities...");

executing the code on a processor (col. 26, line 61, "Machine code is executed…");

detecting the execution event (col. 5, line 39-43, "… sampling the program counter when event counters overflow…");

and backtracking a displacement from a detection point in the code coinciding with the detection of the execution event to a preceding operation associated with the execution event, the backtracking identifying the preceding operation, determining whether the ambiguity creating location is not encountered while backtracking; when the ambiguity creating location is not encountered, associating the execution event with the preceding operation;and wherein the displacement is based, at least in part, on a type of operation appropriate to have triggered the execution event(col. 1, line 35-40, "… to monitor specific events during execution of a program…"; col. 3, 35-49, "for example a load instruction… static analysis… can work backwards… to identify the actual instruction that caused the event.")

As per claim 3, the rejection of claim 1 is incorporated; further Anderson et al. disclose **when the ambiguity creating location is encountered while backtracking, either ignoring the execution event or bridging the ambiguity creating location**(col. 25, line 17-22, "For each instruction executed in the trace, work backwards to determine path segments until either: (a) the global branch history bits are exhausted…").

As per claim 5, the rejection of claim 1 is incorporated; further Anderson et al. disclose **the ambiguity creating location is one of: a jump target location; an indirect branch target location; a branch target location; entry point location; a trap handler location; and an interrupt handler location** (col. 23, line 56-58, "asynchronous events that cause branched code to execute… such as interrupts… can pollute branch history…").

As per claim 6, the rejection of claim 1 is incorporated; further Anderson et al. disclose **the preceding operation corresponds to a load instruction** (;
and the execution event is a cache miss (col. 12, line 31, "Events could include cache misses…").

As per claim 7, the rejection of claim 1 is incorporated; further Anderson et al. disclose **the preceding operation corresponds to a memory access instruction** (col. 9, line 49-50, "… execute memory access instructions…");
the execution event is either a hit or a miss at a level in a memory hierarchy (col. 12, line 31, "Events could include cache misses…").

As per claim 8, the rejection of claim 1 is incorporated; further Anderson et al. disclose **the execution event is either an overflow or an underflow of a hardware counter** (col. 5, line 38-40, "… sampling the program counter when event counter overflow…"; col. 6, line 62-65, "… hardware event-counters…").

As per claim 9, the rejection of claim 1 is incorporated; further Anderson et al. disclose **the execution event triggers either an overflow or an underflow of a hardware counter that is itself detected** (col. 12, line 31-35, "Events could include cache misses…"; col. 5, line 38-40, "… sampling the program counter when event counter overflow…"; col. 6, line 8, "Special-purpose hardware could count…").

As per claim 10, the rejection of claim 1 is incorporated; further Anderson et al. disclose **the latency includes that associated with delivery of a trap** (col. 9, line 59-65, "…some instructions may abort or be trapped. For example, the execution flow may change after an instruction is fetched, or an instruction may suffer an exception trap. In these cases, the instruction and all subsequent instructions already in the pipeline are discarded and the speculative processing state is rolled back." The examiner asserts that the action to discard or rollback instructions takes time and inherently introduces latency.)

As per claim 11, the rejection of claim 1 is incorporated; further Anderson et al. disclose **the latency includes that associated with delivery of a counter overflow event signal** (col. 12, line 51-68, "Latency registers store timing information taken at check points… The checkpoints may differ from processor to processor depending on where an instruction might be stalled waiting for some event or resource. Each latency register counts the number of cycles an instruction spent between two checkpoints." The examiner interprets that one processor may use a counter overflow event as check point for latency.)

As per claim 12, the rejection of claim 1 is incorporated; further Anderson et al. disclose **the latency is associated with pipeline execution skid** (col. 12, line 56-58, "… latency regist4r counts the number of cycles… between two checkpoints.").

As per claim 13, the rejection of claim 1 is incorporated; further Anderson et al. disclose **the latency is associated with completion of in-flight operation** (col. 12, line 51-52, "Latency registers stores timing information… while a selected instruction is in flight".)

As per claim 14, the rejection of claim 1 is incorporated; further Anderson et al. disclose **embodied in a computer program product** (col. 8, line 22-30, "… instructions and data of software programs are stored in the memories… instructions are decoded for execution.")

As per claim 15, the rejection of claim 1 is incorporated; further Anderson et al. disclose **embodied in at least one of: a profiling tool; a code optimizer; and a runtime library** (col. 10, line 13-21, "This makes the profile record useful… profile-directed optimization…").

As per claim 19, Anderson et al. disclose a method of identifying operations associated with execution events of a processor, the method comprising:

During machine code generation of the execution sequence of operations (c23: 24-31, "Many compiler optimizations …either through instrumentation or simulation…"), Indenfying a target operation and an ambiguity creating location within an execution sequence of operation of the processor (col. 1, 35-40, shows using instrumentation for profiling code; col. 23, line 55-col. 24, line 31, for example, col. 24, line 9, "…ambiguities…");

from a point in an execution sequence of the operations of the processor, the point coinciding with an execution event, backtracking through the operations toward a particular operation that precedes the coinciding point by a displacement, wherein the displacement is based, at least in part on a type of operation appropriate for triggering the execution event; determining whether the ambiguity creating location is encountered while backtracking; when the ambiguity creating location is not encountered, associating the execution event with the particular operation. (col. 3, 35-49, "for example a load instruction… static analysis… can work backwards… to identify the actual instruction that caused the event."; col. 7, line 7-9, "Latencies of instructions of the program are measured…".);

As per claim 20, the rejection of claim 19 is incorporated; further Anderson et al. disclose executing **the sequence of operations on the processor** (col. 26, line 61, "Machine code is executed…");

and detecting the execution event (col. 5, line 39-40, "… sampling the program counter when event counters overflow…").

As per claim 21, the rejection of claim 19 is incorporated; further Anderson et al. disclose **the operations are instructions executable on the processor** (col. 2 line 12-13, "… operation of a processor that can issue instructions…");
and the particular operation is a particular one of the instructions that triggers the execution event (col. 16, line 13-18, "As the instruction progresses… trigger signals… Ptrap… ").

As per claim 22, the rejection of claim 19 is incorporated; further Anderson et al. disclose **the operations correspond to instructions of program code.** (col. 2 line 12-13, "… operation of a processor that can issue instructions…").

As per claim 23, the rejection of claim 19 is incorporated; further Anderson et al. disclose **the execution event is an exception triggering execution of the particular operation** (col. 12, line 31-38, "Events could include…. And exception conditions…").

As per claim 24, the rejection of claim 19 is incorporated; further Anderson et al. disclose **the execution event is a cache miss** (col. 12, line 31, "Events could include cache misses…").

As per claim 25, the rejection of claim 19 is incorporated. It recites the same limitation as claim 10 and is rejected for the same reason set forth in connection with the rejection of claim 10 above.

As per claim 26, the rejection of claim 19 is incorporated; further Anderson et al. disclose **the execution event triggers a hardware event and the displacement is based, at least in part, on delivery of a signal associated therewith** (col. 12, line 51-68, "Latency registers store timing information taken at check points… The checkpoints may differ from processor to processor depending on where an instruction might be stalled waiting for some event or resource. Each latency register counts the number of cycles an instruction spent between two checkpoints." The examiner interprets that one processor may use a counter overflow event as check point for latency.)

As per claim 27, the rejection of claim 26 is incorporated; further Anderson et al. disclose **the hardware event is either underflow or overflow of a counter associated with the execution event** (col. 5, line 38-40, "… sampling the program counter when event counter overflow…"; col. 6, line 62-65, "… hardware event-counters…").

As per claim 28, the rejection of claim 19 is incorporated; further Anderson et al. disclose **the execution event is either underflow or overflow of a counter** (col. 5, line 38-40, "… sampling the program counter when event counter overflow…").

As per claim 29, the rejection of claim 19 is incorporated; further Anderson et al. disclose **instances of intervening control transfer targets are identified in the execution sequence of operations to facilitate the backtracking** (col. 24, line 26-31, "… perform a backward analysis… can identify execution paths…").

As per claim 30, the rejection of claim 29 is incorporated; further Anderson et al. disclose **at least some of the instances of intervening control transfer targets are resolved using branch history information** (col. 25, line 17-22, "For each instruction executed in the trace, work backwards to determine path segments until either:

(a) the global branch history bits are exhausted…").

As per claim 31, the rejection of claim 19 is incorporated; further Anderson et al. disclose **an ambiguity creating location in the execution sequence are identified by a compiler** (col. 24, line 33-34, "The analysis can identify execution Paths…").

As per claim 37, the rejection of claim 19 is incorporated; further Anderson et al. disclose **the particular operation comprises a memory referencing instructions** (col. 12, line 13-15, "… the instruction is a memory access instruction…");

As per claim 38, the rejection of claim 37 is incorporated; further Anderson et al. disclose **the memory referencing instruction comprises one or more of loads, stores and prefetches** (col. 12, line 13-15, "… the instruction is a memory access instruction, such as a load…");

As per claim 46, Anderson et al. disclose a method of preparing code, the method comprising: **preparing a tangible first executable instance of the code**(col. 8, line 22-25, "During operation of the system, instructions and data of software programs are stored in the memories. The instructions and data are generated conventionally using known compiler…" It is inherent that a compiler prepares the execution sequence of the operations.),

the preparing identifying at least ambiguity creating locations therein(col. 24, line 33-34, "The analysis can identify execution Paths…");

executing the first executable instance and responsive to detection of an execution event(col. 26, line 61, "Machine code is executed…"; col. 1, line 35-40, "… to monitor specific events during execution of a program…");

backtracking a displacement through the code to an operation thereof, wherein the displacement is based, at least in part, on a type of operation appropriate to have triggered the execution event; and associating the operation with the execution event.(col. 3, 35-49, "for example a load instruction… static analysis… can work backwards… to identify the actual instruction that caused the event.";);

and associating the execution event with one or more of the ambiguity creating locations associated when one or more of the ambiguity creating locations are

encountered while backtracking (col. 24, lines 26-38, "…analysis can identify execution path…branch history…").

As per claim 47, the rejection of claim 46 is incorporated; further Anderson et al. disclose **the association between the associated operation and the execution characteristic is based on a set of additional detections and responsive backtracking** (col. 25, line 17-22, "For each instruction executed in the trace, work backwards to determine path segments until either: (a) the global branch history bits are exhausted…").

As per claim 49, the rejection of claim 46 is incorporated; further Anderson et al. disclose **resolving at least some intervening ones of the identified ambiguity creating locations using branch history information** (col. 25, line 17-22, "For each instruction executed in the trace, work backwards to determine path segments until either:

(a) the global branch history bits are exhausted…").

As per claim 59, Anderson et al. disclose

Means for identifying an ambiguity creating location(col. 23, line 55-col. 24, line 31, for example, col. 24, line 9, "…ambiguities…");

means for backtracking predetermined displacement from a point coinciding with an execution event in an execution sequence of operations on a processor, through the execution sequence toward a particular operation thereof that precedes the coinciding point, wherein the backtracking displacement is based, at least in part, on a tpe of the particular operation;

means for determining whether the ambiguity creating location is not encountered during the backtracking;

and means for associating the execution event with the particular operation when the ambiguity creating location is not encountered. (col. 1, line 35-40, "… to monitor specific events during execution of a program…"; col. 3, line 46-49, "… static analysis… can work backwards… to identify the actual instruction that caused the event.").

As per claim 60, the rejection of claim 59 is incorporated; further Anderson et al. disclose **means for bridging at least some ambiguity creating locations** (col. 25, line 17-22, "For each instruction executed in the trace, work backwards to determine path segments until either: (a) the global branch history bits are exhausted…").

As per claim 62, the rejection of claim 3 is incorporated; further Anderson et al. disclose wherein the ambiguity creating location is bridged using branch history information (col. 25, line 17-22, "For each instruction executed in the trace, work backwards to determine path segments until either: (a) the global branch history bits are exhausted…").

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 16-18 rejected under 35 U.S.C. 103(a) as being unpatentable over Anderson et al. (US Patent No. 5,964,867) in view of Ronstrom (US PGPub. No. 2002/0010913).

As per claim 16, the rejection of claim 1 is incorporated.

Anderson et al. do not disclose padding operations.

However, Ronstrom discloses **a compiler that pads the code with one or more padding operations to absorb at least some instances of the latency** (p. 1, [0012], "… the compiler… generating a dummy instruction code for lowering cache miss penalty and inserting the same…").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teaching of Ronstrom into the teaching **of** Anderson et al. to include padding dummy instruction. The modification would be obvious to one of ordinary skill in the art to want to lower cache miss penalty (p. 1, [0012], line 12-13, "…lowering cache miss penalty…")

As per claim 17, the rejection of claim 16 is incorporated; further Ronstrom disclose **the**

padding operations are not themselves associated with the execution event (p. 1, [0012], "… the compiler… generating a dummy instruction… and inserting the same…" It is inherent that a dummy operation does nothing but consumes processor cycles. So, dummy instructions are not associated with the execution events.).

As per claim 18, the rejection of claim 16 is incorporated; further Ronstrom disclose **each padding operation is not an ambiguity creating location** (p. 1, [0012], "… the compiler… generating a dummy instruction… and inserting the same…" A dummy operation does nothing but consumes processor cycles. So, padding dummy instructions does not create ambiguity locations.)

3. Claims 63-68 are rejected under 35 U.S.C. 103(a) as being unpatentable over Anderson et al. (US Patent No. 5,964,867) in view of Bala et al. ("Efficient Instruction Scheduling Using Finite State Automata").

As per claim 63,
the rejection of claim 1 is incorporated;
further Anderson et al. disclose

- identifying the preceding operation (col. 3, 35-49, "for example a load instruction… static analysis… can work backwards… to identify the actual instruction that caused the event.").

- wherein the expected detection latency is based, at least in part, on a type of the identified preceding operation (col. 3, 35-49, "for example a load instruction… static analysis… can work backwards… to identify the actual instruction that caused the event.")

Anderson et al. do not specifically disclose

- identifying an ambiguity creating location, determining an interval and inserting padding operations.

However, Bala et al. disclose

- identifying an ambiguity creating location subsequent to the identified preceding operation (p. 47, left col., 3rd para., "…the ability to insert an instruction into the middle of an already scheduled block…scheduled above a branch…" The examiner asserts that a branch is an ambiguity creating location and it is inherent to have to a step of identifying such location before inserting instruction before it.).,
- determining that an unambiguous interval between the ambiguity creating location and the identified operation is insufficient to cover an expected detection latency for the identified preceding operation; and inserting padding operations, which provide at least a portion of a skid region between the identified preceding operation and the ambiguity creating location, into the code subsequent to the identified operation, (p. 46, left col., section 1, line 6-8, "…insert explicit NOP instructions in the stream to prevent hazards…"; p. 47, left col., 3rd para., "…the ability to insert an instruction into the middle of an already

scheduled block…scheduled above a branch…" The examiner asserts that such insertion of instruction is a result of determining that an unambiguous interval between the ambiguity creating location and the identified operation is insufficient to cover an expected detection latency for the identified preceding operation, therefore such determination is inherent).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Bala et al. into the teachings of Anderson et al. to include identifying an ambiguity creating location subsequent to the identified preceding operation determining that an unambiguous interval between the ambiguity creating location and the identified operation is insufficient to cover an expected detection latency for the identified preceding operation; and inserting padding operations, which provide at least a portion of a skid region between the identified preceding operation and the ambiguity creating location, into the code subsequent to the identified operation. The modification would be obvious to one of ordinary skill in the art to want to effectively detect hazards as suggested by Bala et al. (p. 47, left col., line 4-5, "…effectively reducing the problem of structural hazard detection…")

As per claim 64,

the rejection of claim 19 is incorporated; see reason for rejection of claim 63 of identifying.

As per claim 65, the rejection of claim 64 is incorporated;

further Anderson et al. disclose

- ignoring a second execution event when the ambiguity creating location is encountered while backtracking (col. 23, line 64, "…events can be ignored…").

As per claim 66, the rejection of claim 64 is incorporated; further Anderson et al. disclose

- encountering the ambiguity creating location while backtracking; and bridging the ambiguity creating location using branch history information (col. 25, line 17-22, "For each instruction executed in the trace, work backwards to determine path segments until either: (a) the global branch history bits are exhausted…").

As per claim 68, the rejection of claim 64 is incorporated;

Anderson et al. do not specifically disclose inserting padding operations.

However, Bala et al. disclose

- inserting one or more padding operations into the sequence of operations between the identified target operation and the ambiguity creating location to cover an expected detection latency of the target operation (p. 46, left col., section 1, line 6-8, "…insert explicit NOP instructions in the stream to prevent hazards…"; p. 47, left col., 3rd para., "…the ability to insert an instruction into the middle of an already scheduled block…scheduled above a branch…").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Bala et al. into the teachings of Anderson et al. to include inserting one or more padding operations into the sequence of operations between a first of the identified target operations and a first of the identified ambiguity creating locations to cover an expected detection latency of the first identified target operation. The modification would be obvious to one of ordinary skill in the art to want to effectively detect hazards as suggested by Bala et al. (p. 47, left col., line 4-5, "…effectively reducing the problem of structural hazard detection…")

As per claim 69, the rejection of claim 46 is incorporated;

Anderson et al. do not specifically disclose

- inserting padding operation to provide a skid region.

However, Bala et al. dislcose

- inserting one or more padding operations between the operation and a first of the identified ambiguity creating locations that is subsequent to the operation, wherein the one or more padding operations provide at least a portion of a skid region between the operation and the first identified ambiguity creating location sufficient to cover an expected detection latency of the operation (p. 46, left col., section 1, line 6-8, "…insert explicit NOP instructions in the stream to prevent hazards…"; p. 47, left col., 3rd para., "…the ability to insert an instruction into the middle of an already scheduled block…scheduled above a branch…".)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Bala et al. into the teachings of Anderson et al. to include inserting padding operation to provide a skid region.

The modification would be obvious to one of ordinary skill in the art to want to effectively detect hazards as suggested by Bala et al. (p. 47, left col., line 4-5, "…effectively reducing the problem of structural hazard detection…")

4. Claims 50-54, and 61 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ronstrom (US PGPub. No. 2002/0010913) in view of Bala et al. ("Efficient Instruction Scheduling Using Finite State Automata").

As per claim 50, Ronstrom discloses a computer program product encoded in one or more computer readable media, the computer program product (p. 3, [0037], line 2-3, "… a computer program product…loadable into the memory..") comprising:

an execution sequence of operations (p. 4, [0063], line 3-4, "… a machine code running on a standard processor…");

Ronstrom does not specifically disclose

- Inserting padding instruction to provide a skid region.

However, Bala et al. disclose

- **one or more padding operations following a first operation of the execution sequence, the padding operations providing an unambiguous skid region**

> **of the execution sequence between the first operation and a subsequent ambiguity creating location to provide a displacement between the subsequent ambiguity location and the first operation to cover an expected detection latency of the first operation** (p. 46, left col., section 1, line 6-8, "…insert explicit NOP instructions in the stream to prevent hazards…"; p. 47, left col., 3rd para.,; p. 47, 3rd para., line 6-10, "…insert instruction…when an instruction is speculatively scheduled above a branch…" The examiner asserts that a branch is an ambiguity creating location.).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Bala et al. into the teachings of Ronstrom to include padding instructions between some particular operations and the ambiguity creating locations to create a skid region. The modification would be obvious to one of ordinary skill in the art to want to effectively detect and reduce hazards as suggested by Bala et al. (p. 47, left col., line 4-5, "…effectively reducing the problem of structural hazard detection…")

As per claim 51, the rejection of claim 50 is incorporated; further Ronstrom discloses **the first operation includes a memory access operation** (p. 1, [0012], line 9-13, "… detecting cache miss penalty… generating dummy instruction code for lowering cache miss penalty and inserting the same…" The examiner asserts that operations related to cache miss are memory access operations.)

As per claim 52, the rejection of claim 50 is incorporated;

further Bala et al. discloses **the padding operations include nops** (p. 46, section 1, line 7, "…insert explicit NOP instruction…")

As per claim 53, the rejection of claim 50 is incorporated; further Ronstrom discloses **the unambiguous skid region does not include an ambiguity creating location** ((p. 46, section 1, line 7, "…insert explicit NOP instruction…" The examiner asserts that inserted NOPS does not create ambiguity.)

As per claim 54, the rejection of claim 50 is incorporated; further Ronstrom discloses **the one or more computer readable media are selected from the set of a disk, tape, magnetic, optical, semiconductor or electronic storage medium** (p. 3, [0039], line 7-8, "… floppy discs and hard drives…" The examiner asserts that a hard drive is a magnetic storage medium.)

As per claim 61, it is the apparatus claim that recites the same limitation as claim 50 and is rejection for the same reason set forth for the rejection of claim 50.

Response to Arguments

In the remark,

Applicant argues:

1) Anderson Does Not Disclose Backtracking A Displacement From A Detection Point Based On The Type Of Operation Triggering The Execution Event

Examiner's response:

1) Upon reviewing the Applicant's specification, [1030] (or similarly [1038]) --

"Beginning at the coinciding point for a particular execution event detection, collection steps back through the original execution sequence in an attempt to find a preceding target instruction (i.e., an instruction of appropriate type to have triggered the detected event). In some realizations, an expected displacement is reasonably fixed and may place an upper bound on the backtracking."

The above disclosure appears to suggest the backtracking attempts to find a preceding target instruction of appropriate type to have triggered the detected event. That means the type of instruction that has triggered the detected event is yet to be determined. However, the claim language appears to be able to determine a displacement for backtracking based on a type that appears to be unknown when such determination is made. Further the specification appears to suggest an expected displacement is reasonably fixed or there is an upper bound on the backtracking. It does not appear to disclose any determination of displacement based on a type of operation.

Applicant argues:

2) B. Anderson Does Not Disclose Identifying Ambiguity Creating Locations During Machine Code Generation

Examiner's response:

2) Per Anderson, column, lines 24-31, "Many compiler optimizations…either through instrumentation…" where Anderson discloses using a compiler to trace execution paths. Since compiler is used, therefore it discloses the limitation of during machine code generation.

Applicant argues:

3) Bala does not disclose or suggest "padding the execution sequence to provide an unambiguous skid region between a particular operation and a subsequent ambiguity creating location within the sequence of operations to cover an expected detection latency" as required by independent claims 50 and 61.

Bala teaches techniques for avoiding pipeline hazards when an instruction is speculatively executed above a branch upon which it is control dependent. *See Bala, page 46, section 1, paragraph 1, lines 1-8 and page 47, section 1, last paragraph.* That is, Bala discloses techniques for relocating instructions above a branch that are not dependent upon the branch being taken or not taken such that the reordering of instructions preserve control dependences. In contrast, the independent claims require padding the execution sequence to provide an unambiguous skid region between a particular operation and a subsequent ambiguity creation location within the sequence of operations to cover an expected detection latency.

Examiner's response:

3) In Bala, p. 46, left col., section 1, line 6-8, "…insert explicit NOP instructions in the stream to prevent hazards…"; According to Bala, the insertion of NOP (or padding the execution

sequence) is sufficient to prevent hazard, that means there are sufficient number of NOP instructions being inserted into the execution sequence to cover an expected detection latency. These inserted NOP instructions contains no branch instruction and therefore is an unambiguous skid region. Any instruction can be after the inserted NOP instruction, including a branch instruction.

Conclusion

THIS ACTION IS MADE FINAL. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Philip Wang whose telephone number is 571-272-5934. The examiner can normally be reached on Mon - Fri 8:00 - 4:00PM. Any inquiry of general nature or relating to the status of this application should be directed to the TC2100 Group receptionist: 571-272-2100.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Wei Zhen/

Supervisory Patent Examiner, Art Unit 2191